Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access

Elke Michlmayr*
Women's Postgraduate College
for Internet Technologies (WIT),
Vienna University of Technology,
Vienna, Austria
michlmayr@wit.tuwien.ac.at

Steve Cayzer
HP Labs,
Filton Road, Stoke Gifford
Bristol BS34 8QZ
United Kingdom
steve.cayzer@hp.com

ABSTRACT

Due to the high popularity of social bookmarking systems, a large amount of metadata is available. Aggregating the metadata belonging to one user results in an user profile similar to those often used in Information Filtering. This paper shows how to create user profiles from tagging data. We present the Add-A-Tag algorithm for profile construction which takes account of the structural and temporal nature of tagging data. In addition, we explore ways of leveraging these user profiles. There are two main insights gained. Firstly, as we experienced in a small-scale user study, simply being able to view aggregated information about past tagging behavior was considered useful. Secondly, the user profile can be used to guide the user's navigation, that is, to provide the user with personalized access to information resources.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

tagging user profiles dynamics information filtering visualisation hci

1. INTRODUCTION

Social bookmarking systems, such as del.icio.us [19], have been around for quite a while now. They provide interfaces for annotating bookmarks with free-text keywords. Their simplicity and their immediate usefulness for improved rediscovery of information have attracted a high number of users. All users' annotated bookmarks are by default publicly accessible. Hence, an immense amount of metadata is available. This collaboratively created data is a valuable resource. Aggregations of it are provided to the user community. Several papers address analysis [9] or data mining [12] of tagging data. Most authors analyse the properties of the metadata related to certain bookmarks and/or to certain tags. In this paper, we focus on those tags which have been employed by a certain user. We treat them as a continuous stream of information about a user's interests, which can be used for creating a rich user profile.

Aggregated information about a user's bookmark collection is usually represented as a tag cloud, in which all tags a user has employed so far are listed alphabetically and the font size of a tag is set according to how often it has been used so far. Our claim is that tag clouds fail to represent two important properties of a user's bookmark collection.

- Firstly, they do not represent the relationships between the tags, which can be derived by using co-occurrence techniques.
- Secondly, they do not consider that tagging data is time-based in their weighting of the relative importance of a tag.

Our aim is to learn user profiles from tagging data that include those two properties. In addition to creating profiles, we need to present them to the user in such a way that it serves him or her a useful purpose. One such purpose is being able to view the structure and the contents of the profile to get an overview of a user's interests. But other than that – just as tag clouds are used as an interface to access a user's bookmark collection – aggregated information about tagging behavior can also be exploited to provide a user interface to browse *some other source of data* through an representation of a user profile.

This paper is organized as follows. In Section 2 we discuss various possibilities for creating a user profile out of a tag collection. This provides the design rationale for the Add-A-Tag algorithm, which is formally defined in Section 3. Section 4 presents user profile visualisation methods together with user feedback about them. The emphasis is put on the visualising the profile's changes over time. Section 5 shows how users can browse information resources through an representation of the user profile. Finally, Section 6 gives an overview of related work and Section 7 concludes.

2. PROFILE CONSTRUCTION

In this section we present three different methods for profile construction. Section 2.1 describes the naive approach, Section 2.2 the co-occurrence approach, and Section 2.3 the adaptive approach. Examples are used to illustrate the approaches, the sample data for which are shown in Figure 1.

*This research has partly been funded by the Austrian Federal Ministry for Education, Science, and Culture (bm:bwk), and the European Social Fund (ESF) under grant 31.963/46-VII/9/2002.

WWW2007, May 8–12, 2007, Banff, Canada.
.

```
1  datamining rdf tools web
2  algorithms design geo java library programming
3  danger security pc tools web
4  ais security research article
5  bbc media rss social syndication
6  blog flickr fun geo metadata social uk web
7  ai turing teaching
8  ajax eclipse programming jsp spring tools uml web
9  geo google gps javascript tools web web2.0
10 owl rdf semanticweb web2.0
11 ai teaching
12 ai teaching
13 teaching ai
14 ontology opensource research security
15 design research robot ai teaching
```

Figure 1: Sample data. A user stores a collection of 15 bookmarks. These bookmarks are annotated with the tags shown as space-separated lists. The lists are ordered according to the time the corresponding bookmarks were added to the bookmark collection. The oldest one is shown first (line 1). Note that this is a very small data sample, for explanatory purposes.

Consider a user's bookmark collection consisting of a user-defined number of bookmarks. Each bookmark in the collection is composed of a title, a description, a URL, a date, and a set of tags. For creating the profile, we focus on the tags and their temporal ordering by increasing date.

2.1 Naive approach

To construct a user profile out of these data, the task is to aggregate it in such a way that the interests of the user are reflected according to their intensity. The more often a certain tag is used, the higher the interest of the user in the corresponding topic. Therefore, the most simple method for creating aggregated data for a user's bookmark collection is to count the occurrence of tags. This is the approach taken for creating tag clouds. The result of this computation is a list of tags which is ranked according to tag popularity. For the sample data (Figure 1), the ranked tag list is shown in Table 1. It reveals that most tags have been used only once, and that there are only a few tags which were used often. The user profile can then be created by selecting the top k most popular tags from the ranked list. If we select the top 3 tags, for example, the resulting user profile consists of the tags: `web ai teaching`. The benefit of this method is that it is very simple, and hence fast. However, it has some drawbacks. One major problem is that those tags which are most often used tend to be not very specific (e.g., the tag `web` is a very general one). Moreover, the resulting profile consists of unlinked tags. Although the tagging data includes information about the relationships between those tags, these relationships are not included in the user profile. The co-occurrence approach presented in the next section tackles both these drawbacks.

#Occ.	Tag
5	web, ai, teaching
4	tools
3	security, research, geo
2	web2.0, rdf, social, programming, design
1	semanticweb, danger, rss, turing, metadata, jsp, fun, library, owl, article, ontology, google, eclipse, ajax, syndication, ais, javascript, bbc, robot, media, pc, uml, flickr, blog, java, spring, datamining, gps, opensource, uk, algorithms

Table 1: List of tags ranked by their number of occurrence

2.2 Co-occurrence approach

The resulting profile is more specific if we focus not only on which tags have been used, but rather on which tags have been used in combination. This can be achieved by relying on the co-occurrence technique known from Social Network Analysis [17]. If two tags are used in combination (*co-occur*) by a certain user for annotating a certain bookmark, there is some kind of semantic relationship between them. The more often two tags are used in combination, the more intense this relationship is. This is represented by a graph with labeled nodes and undirected weighted edges in which nodes correspond to tags and edges correspond to the relationship between tags. Each time a new tag is used, a new node for this tag is added to the graph. Each time a new combination of tags is used, a new edge with weight 1 between the corresponding nodes is created in the graph. If two tags co-occur again, the weight for the corresponding edge is increased by 1.

The graph is created by parsing the tags for all items in the bookmark collection and applying the technique described above. In the second step, a user profile is derived from the resulting graph by selecting the top k edges with the highest weights and their incident nodes. Figure 2 shows the resulting graph when applying the co-occurrence approach to the sample data. A ranked list of the weights of the resulting graph's edges for the sample data is shown in Table 2. Selecting the top 3 edges and their incident nodes for the user profile returns a graph with 5 nodes and the following edges: `ai-teaching tools-web geo-web`.

Co-occurrence techniques have been employed for diverse purposes. First and foremost, the folksonomy providers rely

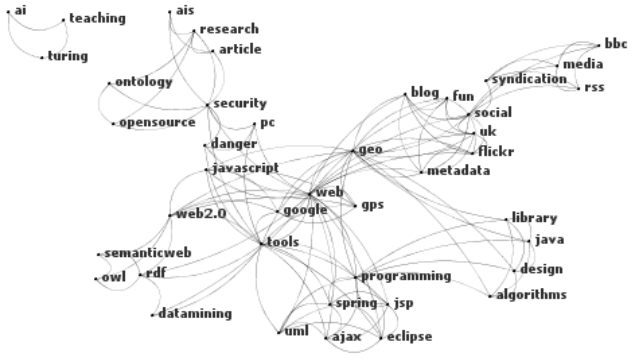

Figure 2: Co-occurrence network for the sample data shown in Figure 1. Two nodes are linked with an edge if the corresponding tags have been used in combination for annotating a bookmark. Edge weights are not shown. Note that although the amount of sample data is rather small, the resulting network is quite big.

Weight	Tag combination
4	ai - teaching, tools - web
2	geo - web, security - research

Table 2: Top 4 tag combinations ranked by their number of occurrence using the co-occurence technique (Section 2.2)

on it for computing related tags. Moreover, co-occurrence is also used in knowledge discovery from databases [3], for extracting light-weight ontologies from tagging data [12], or for tag recommendation [2, 18].

The novelty of our approach is that we use co-occurrence at a smaller scale: for one bookmark collection, only. The impact of this is that the relationships between the tags are not the result of a community-driven process, but entirely created by one user instead. Hence, the relationships between the tags might not make sense to anyone except to the user who created them. However, in the case of user profile creation this is acceptable and even desirable, because for this task we need to find out about how the interests of a user are connected to each other, no matter how unorthodox these connections might be.

One drawback of the co-occurrence approach is that it does not include bookmarks that are annotated with a single tag. In order to overcome this issue, it would be necessary to combine it with the naive approach. The result would be a graph with weighted nodes and weighted edges. However, we decided against a combination of approaches, because the average percentage of bookmarks annotated with only one tag by our user population (Section 4.1) is 8%. This can serve as an indicator that the average percentage of bookmarks annotated with only one tag on del.icio.us is small. Therefore, we accept the loss of these data in favour of a simpler method. Another drawback of this approach is that the age of bookmarks and their temporal ordering is not considered. This issue is addressed by the adaptive approach presented in the next section.

2.3 Adaptive approach

Since social bookmarking systems have been around for quite a while now, many of their users manage a rather big bookmark collection which they continuously have been adding items to for the time span of several months or even years. The average lifetime of the bookmark collections of the users that participated in the user study (Section 4.1) is 607.7 days. Therefore, the age information of the tagging data is important. It makes a difference if a user has used a certain tag and, therefore, specified a certain interest, one day or one year ago. To include the age of the bookmarks in the user profile we extend the co-occurrence approach with the evaporation technique known from ant algorithms [5]. Evaporation is a simple method to add time-based information to the weights of edges in a graph: Each time the profile graph is updated with tags from a newly added bookmark, the weight of each edge in the graph is decreased slightly by removing a small percentage of its current value.

Obviously, when creating the profile graph for the adaptive approach by parsing the tags for all items in the bookmark collection, it is necessary to start parsing from the oldest item and to process the items in the same temporal order as they were added to the bookmark collection. Again,

Weight	Tag combination
3.83	ai - teaching
3.63	tools - web
1.89	security - research
1.85	geo - web

Table 3: Top 4 tag combinations for the adaptive approach with parameters $\alpha = 1.0, \beta = 1.0, \rho = 0.01$ (see Section 2.3 for details).

the user profile is created by extracting the top k edges with the highest weights and their incident nodes from the profile graph. Applying the adaptive approach to the sample data apparently returns the same profile graph as before (Figure 2), but with different weights of the links in this graph. Table 3 lists the highest weighted edges in this graph. Selecting the top 3 edges and their incident nodes for the user profile returns a graph with 6 nodes and the following edges: `ai-teaching tools-web security-research`. Note that the combinations `geo-web` and `security-research` occur the same number of times in the sample data. In the co-occurrence approach, the weight was the same for both combinations and therefore it was necessary to randomly select one of them for the profile. With the adaptive approach it is possible to detect that the latter combination has been used at a later point in time and can therefore be considered as currently more important to the user.

3. THE Add-A-Tag ALGORITHM

Now we formally define the adaptive algorithm that was described in Section 2.3. Consider a user u adding a bookmark item b tagged with tags $t_1, \ldots, t_n$ to his or her bookmark collection. The profile graph $G_u = (V, E)$, where $V = v_1, \ldots, v_n$ is the set of vertices (which correspond to tags), and $E = e_1, \ldots, e_n$ is the set of edges, is updated as follows.

Evaporation In the first step, the existing information in the graph is changed by applying the evaporation formula shown in Equation 1 to every edge $e_x \in E$

$$w_{e_x} \leftarrow w_{e_x} - \rho \cdot w_{e_x} \ , \tag{1}$$

where $\rho \in [0, 1]$ is a constant and w_{e_x} is the weight of edge e_x. The value used for ρ defines the relative importance of the most recently used tags. The higher the value for ρ, the more emphasis is put on them.

Reinforcement In the second step, the n new tags from bookmark b: $t_1, \ldots, t_n$ are added to the graph. For every combination $t_i t_j$ where $i, j \in 1, \ldots, n$ and $i < j$, the following procedure is executed:

1. For every tag t_x ($x \in i, j$), add a corresponding vertex v_x to graph G_u, if v_x does not exist.
2. If it does not yet exist, add an edge with weight α between vertex v_i and vertex v_j to graph G_u, where constant α is a real number and $\alpha > 0$.
3. Otherwise, if an edge between vertex v_i and vertex v_j exists, increase its weight by β. Constant β is a real number and $\beta > 0$.

The procedure described above is executed each time the user adds an bookmark item to the bookmark collection.

Extracting the user profile from the profile graph is defined as follows.

1. Create a ordered set E_s from $E = e_1, \ldots, e_n$. E_s contains all edges e_x ($x \in 1, \ldots, n$) from graph G_u ordered in decreasing order by their weights w_{e_x}.

2. Create set E_k by extracting the top k elements from set E_s, where k is a natural number and $k > 0$.

3. Create graph $G_{u'}$ which contains all edges from E_k and all vertices from graph G_u which are incident to one of the edges in E_k.

The size of the user profile $G_{u'}$ is determined by the value chosen for parameter k.

4. PROFILE DYNAMICS

Now we need a way to observe the changes in a user profile over time. The visualisation method described in this section was first intended as a "debug tool" to view the creation process of a profile in the design phase of the Add-A-Tag algorithm. However, it turned out to be of high interest to the del.icio.us users among our peers to be able to view their tagging activities in the past. For this reason, it developed into a fully functional tool.

We decided for a graphical representation of the profile instead of a text-based one. This makes it much easier to show the network structure of the profile. To provide for intuitive observation of the dynamic changes, all nodes are moving using a "bubbling up" metaphor, which means that they enter the screen from the bottom and continuously move towards the top. If a tag is included in the user profile at one point of time, but not included in the next state, it vanishes from the screen.

Using this metaphor, visualising the profiles created with the naive approach is straightforward. The nodes are shown as dots and labeled with their corresponding tags. They enter the screen from the bottom on a randomly chosen horizontal position, and bubble up. However, for the co-occurrence approach and for the Add-A-Tag approach, it is also necessary to visualise the edges between the nodes. The lengths of the edges between the nodes need to correspond to the edge's weights. The higher the weight, the shorter the length of the edge must be.

Basically, there are two approaches possible for visualising these dynamic graphs. In the first approach, all nodes and edges that will be included in the profile at a certain point in time need to be known in advance. In the next step, a graph layout algorithm can be applied for calculating the positions of all the nodes and edges. During the animation, those nodes that are currently included in the profile are set to visible while all the others are set to invisible. The benefit of this approach is that the nodes do not move. However, the drawback is that the layout algorithm creates a visually pleasing layout for the complete graph, but the layouts of the different graph states shown over time are not optimized and tend to look quite ugly.

Therefore, we had to adopt another approach by using an iteration-based graph visualisation algorithm that incrementally optimizes the layout of the different graph states. We

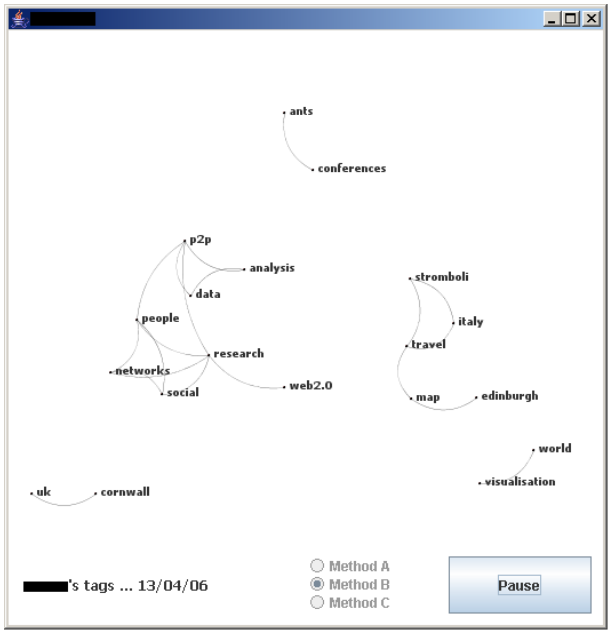

Figure 3: Visualisation of a user profile

chose to combine nodes "bubbling up" with the spring embedder layout algorithm by Fruchterman and Reingold [8], in which the nodes repel or attract each other depending on the edges between them and on the weight of these edges. In addition, a minimum and a maximum length for the edges was defined in order to prevent node labels being printed on top of each other, and to avoid nodes being too far away from each other. The "bubbling up" metaphor and spring embedding work together. If a tag A that newly appears at the bottom of the screen has a connection to a tag B that is already shown on the screen, the spring embedder algorithm will cause tag B to move down on the screen and tag A to move up at the same time. Tag A and tag B will move towards each other until the edge between them has a length according to its weight.

This has desirable impacts on the vertical positions of the profile's components, which divide themselves into *active and not active* as well as into *long-term, mid-term, and short-term* interests of a user. Those subgraphs of the graph which change over time – meaning that new nodes connect to the subgraph – are vertically aligned in the center of the screen (e.g., the two subgraphs related to research and travel in Figure 3), because newly added tags make the older, related tags move down again. They refer to long- and mid-term interests of a user that are currently active. On the contrary, those subgraphs that do not change but are still included in the profile move to the top of the screen (e.g., `ants-conferences` in Figure 3). They refer to long-term interests of a user that are currently not active. The third category are those tags that move in from the bottom and vanish shortly after (e.g., `uk-cornwall` in Figure 3). They refer to short-term interests of a user.

A screen shot of the visualisation tool is shown in Figure 3. The screen is divided into a main part and a control panel at the bottom of the screen. The control panel contains (1) radio buttons which allow the user to select one of the profile creation methods and (2) a button to start the visualisation.

After starting a visualisation, the user profile is presented as an animation over time. The bottom panel shows a date and the main part of the screen shows the state of the user profile at this date. A button allows the user to pause and resume the animation.

4.1 User feedback

In this section we present the results of a small user study conducted in order to get feedback about user's acceptance of the three different profile creation and profile visualisation methods. Six users were provided with the visualisation tool described in the last section. The names of the profile creation methods were not mentioned in order not to influence the results of the user study. In Figure 3, `Method A` refers to the naive approach, `Method B` to the Add-A-Tag approach, and `Method C` to the co-occurrence approach.

They were asked to fill out a questionnaire in which they had to rate the different methods. The following scale was used for the rating: *Very good*, *Good*, *Fair*, *Poor*, *Very Poor*. In addition, the users were asked to rank the methods from 1 to 3 according to how much they liked them, and to justify both the choices for rating and ranking using free-form text. There was also some space for additional comments included. The application and the questionnaire were sent by email to the user, who also replied using email.

As overall feedback, we observed a *Wow!*-effect similar to the one described by Golder et al. [16] in their study of visualising users' email archives. The users were generally pleased with the possibility of viewing aggregated information about their bookmark collection. Both being able to view the (1) relationships between the tags and the (2) trends over time were recognized and appreciated. In their feedback, many of the users mentioned that some tag combinations showed up in the profile at some point of time which they were able to track back to a specific event they could still remember. To cite one of the users: *"I kept having the feeling that by looking at the graph some sort of hidden meaning was coming out. The visualisation style is definitely inspiring, for revealing non-obvious relations!"*

Although the participants in the user study were not provided with any information about the inner working of the different methods, a majority of the users (4 of 6) were able to correctly identify and describe which kind of aggregation was performed for the different approaches, e.g, as one user expressed it: *"I guess method 3 represents the average most used tags, while method 2 the average most recently used tags."*. However, the users' preferences for the different methods turned out to be quite diverse. Two users ranked the co-occurrence approach first, two of them preferred the Add-A-Tag approach, and one of them ranked both of them equally. One of the users favored the naive approach. This may have been down to the visualisation algorithm rather than the profile creation method: *"there was too much movement and too many changes on the screen, and the edges between them were detracting from the tags"*.

The average rating for the naive approach was *Poor*. The average rating for both co-occurrence approach and the Add-A-Tag approach was *Good*. Several users mentioned that they perceived the difference between the co-occurrence approach and the Add-A-Tag approach as being rather small.

We may conclude that the preferred method of user profile creation is a very individual choice. For this reason, instead of creating a tool with a hard coded method, a preferable solution may be to allow the user to choose and configure his or her profile creation algorithm and visualisation method. The popularity of the co-occurrence method shows that users value the long-term tag relationships in their profile; however they also appreciated that Add-A-Tag adapts better to recent changes. Allowing users to select the balance of long-term and short-term interests would provide control without over-burdening the user.

5. PROFILES FOR PERSONALIZED INFORMATION ACCESS

In the following we discuss the usage of the created profile for assisting users in navigating information resources. We present two example scenarios in which the created profile can be of benefit. Section 5.1 discusses the scenario of browsing the Web, Section 5.2 that of an annotated data source. Obviously, the profile can also be used for accessing a user's bookmark collection in the same manner as tag clouds are used for that task.

Since visualising the relationships between the tags and the time-based aspects at the same time would cognitively overload users, in this section we focus on visualising the relationships between the tags in the profile only.

5.1 Browsing the Web

If the person knows what he or she is looking for, e.g., when performing a search, knowing the user's additional interests other from the current one is of minor importance. On the contrary, knowing the user's interests is important if the person does not know what he or she is looking for, e.g, when browsing the Web for no specific purpose. In this case, the profile can be shown in the browser's sidebar or as part of the Web page (similar to a navigation menu). When a tag occurs in the Web page the user is currently looking at, the tag can be highlighted in the profile, and clicking on it results in automatic scrolling to the position on the page on which the tag occurs. Another possibility is to highlight the terms in the Web page that are matched by tags in the profile (e.g., in the same manner as search strings are highlighted when viewing the Google cache of a search result). To improve the recall, string matching is used in combination with stemming.

5.2 Browsing an annotated data source

The situation is more complex if a user wants to access a data source that is annotated with metadata. In this case, a matching needs to be performed. In general, matches are possible between (1) the profile and the content of the data source (as already discussed in the previous section), or between (2) the profile and the metadata of the data source as a description of the corresponding content.

In the following we discuss the latter case using the HP Technical Reports[1] as an example for such a data source. They comprise a document collection annotated with metadata, such as title, author(s), date of publication, number of pages, abstract, and keywords. Only metadata that describe the contents of a resource can be used for the matching. Structural metadata (such as number of pages) is not helpful for matching, but can be exploited for additional navigational options in the interface.

[1]see `http://www.hpl.hp.com/techreports/`

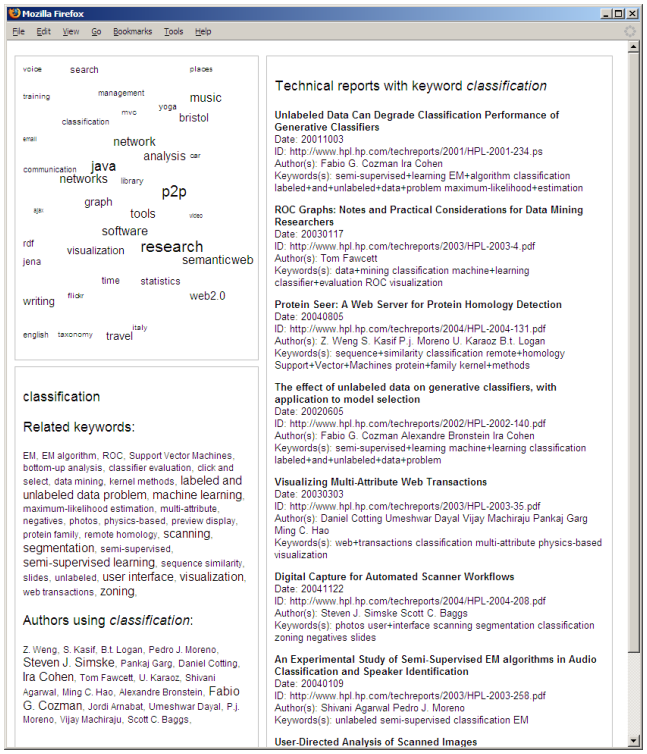

Figure 4: Interface layout. The top left shows the profile. The main screen (right) shows the resources that match with the tag from the profile selected by the user. The bottom left shows additional navigation options.

In our example there are three possibilities for the matching between profile and data source. We can either match (1) tags and keywords, (2) tags and abstracts, or/and (3) tags and full text. If the tags in the profile are from very different domains than the domain of the data source, the matching may not be successful. However, at least a partial overlap between the user's overall interest and his or her current interests can be safely assumed. For the matching itself, string matching in combination with stemming is used. Since tags are most commonly in lower-case letters, whereas keywords are usually in capitalized letters, the matching needs must be performed in a case-insensitive way.

A conceptual overview of the user interface layout is shown in Figure 4. The top left shows a representation of the profile. The user can select a tag from the profile to show only those resources in the main screen on the right that match with the selected tag. The bottom left shows additional navigation options (which are explained later).

The question of how to represent the profile needs to be addressed from two viewpoints. One of them is the *profile-centric viewpoint* which focuses on visualising the structure of the profile. For visualising the relationships between the tags in the profile, a spring embedder layout algorithm is used to position related tags next to each other. For showing the relative importance of a tag, the font sizes are set according to the relative importance of a tag, as in a tag cloud. Figure 4 shows the resulting representation.

However, since the profile will possibly contain tags for which no corresponding data can be found, it is also necessary to take a *data-centric viewpoint* by adapting the profile to the data that is available. Those tags for which no content exists are removed from the profile. For those tags for which corresponding resources exist, an optional possibility would be to print the number of resources that exist next to the tag name, as in faceted browsing. However, we decided against this option because combining font sizes (for relative importance of tags) and numbers (for number of resources) might be misleading to users.

The data source will contain content for which no corresponding tags are included in the profile. Therefore, using only the profile for navigation would make it impossible for the user to access that content. This can be avoided by offering additional navigation options to the user, such as a simple query interface.

Moreover, providing additional context enables improved browsing of the data source. We achieve that using 2 navigation panels, shown in the bottom left of Figure 4. The first shows a list of keywords, each of which co-occurs with the selected keyword. Co-occurrence in this case means that the keywords in question are both attached to a single technical report. These related keywords are likely to cover between them many technical reports, including those which do not have any keyword matching a user's tags. The second navigation panel is similar, but this time shows all authors that have used the selected keyword to markup one or more of their technical reports. Again, the union of all technical reports authored by one of these people is likely to include those that would not be covered by the profile alone. The layout of both these panels is similar; the font size represents the relative importance within the dataset; that is, the number of technical reports tagged with this person or keyword. Unlike the profile pane, co-occurrence patterns are not used to influence the relative positions.

We have also investigated the possibility of representing the user profile as a hierarchy. Such a structure would have advantages of simplicity and familiarity. Multiple inheritance issues (that is, a tag having 2 parents) do not preclude such a representation (a tag would just appear in 2 places in the hierarchy). We adopted an approach loosely similar to the one of Heymann and Garcia-Molina's [10], who use centrality measures to derive a taxonomy from tagging data based on the entirety of a folksonomy's tagging data. Two steps which are executed for every subgraph. Firstly, the node with the highest betweenness centrality is determined as the root node of the tree. Secondly, Prim's algorithm [15] is used for computing the maximum spanning tree based on the weights. However, we have found that this approach is not well suited for profile representation of the type we are interested in. One problem is that the resulting tree can be quite unbalanced, which gives an unsatisfying browsing experience. In addition, nodes that frequently co-occur belong conceptually together and should exist at the same hierarchy level, e.g., the tags "semantic" and "web". The spanning tree approach forces these tags to exist at different levels which is confusing for the user. For these reasons we decided to go for the spring embedder layout style as described above.

6. RELATED WORK

The work which is most directly related to ours is ExpertRank [11] for measuring the expertise of a user in the context

of a certain tag. ExpertRank can be viewed as an approach complementary to ours. Instead of determining all areas of expertise for a given user, it finds users that are knowledgeable in a certain area. Time-based aspects are not considered. Concerning the visualisation of the dynamics of tagging data, the most important piece of work is TagLines [6], which provides a visualisation of the most popular tags over time in Flickr [20]. It takes the entirety of Flickr tags into account. The graph visualisation tools presented in this paper are based on the JUNG framework [13]. The GUESS [1] framework supports visualisation of dynamic graphs with the so-called tweening algorithm. Similar the visualisation tool described in this paper, it creates an animation of the changes over time. This animation can be saved to QuickTime format. A TouchGraph-based visualisation of del.icio.us related tags is provided by Alf Eaton [7].

Several applications for visualising a user's tag collection can be found on the Web. *Extisp.icio.us* [4], also described as "del.icio.us scattering" by its author, is a simple HTML-based visualisation that uses the size of a browser window. Just as for a tag cloud, the size of the tags depends on their popularity. The output looks similar to the one presented in Section 5.2, but unlike as in our approach, the tags are positioned randomly on the screen and the relationships between the tags are not taken into account. Since the tags are not filtered according to their popularity, the output is quite scattered: Some tags are printed in very small font size, and some on top of each other. *Revealicious* [14] provides three different ways for visualising a user's tag collection. One of them, called *SpaceNav*, is a method for graph exploration. Selecting a tag shows all its neighbors in a circle layout. Selecting a neighbor again brings up its neighborhood. The history of clicked tags is shown as a path. For selected tags, it is also shown how often it has been used and to how many other tags it is related. *TagsCloud* is an extended tag cloud in which hovering over a tag brings up related tags in color. *Grouper* does the same, but additionally groups all tag into the categories "most used", "commonly used", and "less used". *Delicious Soup* [21] shows all tags as dots in a two-dimensional grid. The size of a dot roughly corresponds to the number of times the tag has been used. Hovering over a dots shows textual information about how often and since when the tag has been used, together with the number of related tags. In addition, the related dots are highlighted. *Delicious Soup* could perhaps be improved by positioning related dots next to each other in the grid, and by including the tag for a related dot when highlighting it.

7. CONCLUSION AND FUTURE WORK

In this paper we described a technique for building a user profile from a user's tagging behaviour. It does not seem adequate to take account of tag frequency alone; co-occurrence information is needed to make the profiles include all information available. The user's interests and, therefore, the profiles demonstrably adapt over time. We have shown how mechanisms like evaporation should be used to deal with transient and changing interests. Visualisation of the generated profiles is not straightforward but seems genuinely useful. We have described one way to deal with short and long term interests simply and effectively.

Profiles can also be used to access information sources – other than they have been created from – in a way that makes sense to the user. We have shown an example where tag profile information guides access to the HP Labs technical report archive. We would like to scale this up to provide a production ready service in the future.

One issue still unsolved is that of profile representation. We found a simple graph like layout algorithm worked better than tag clouds or hierarchical layout. However, we would like to have larger scale user feedback to support our conclusions. Also the spectrum of layout algorithms is far from exhausted; given the utility of such profiles to the user it seems this would be a promising direction for our work.

The matching of profiles to information sources has to date been achieved using simple mechanisms; string matching in combination with stemming and case conversion. This could be enhanced by backing the comparison algorithm with a thesaurus such as WordNet – this would link tags with synonym keywords, for example. Another possibility would be to use a data-centric approach, such as clustering, to find implicit relationships between tags or technical report keywords. Again, this mechanism would allow a tag to be matched to a larger number of possible keywords.

8. REFERENCES

[1] E. Adar. GUESS: A Language and Interface for Graph Exploration. In *Proceedings of the International Conference on Conference on Human Factors in Computing Systems (CHI2006)*, April 2006.

[2] A. Byde, H. Wan, and S. Cayzer. Personalized Tag Recommendations via Social Network and Content-based Similarity Metrics. In *Proceedings of the International Conference on Conference on Weblogs and Social Media (ICWSM'06)*, March 2006.

[3] H. Chen and K. J. Lynch. Automatic Construction of Networks of Concepts Characterizing Document Databases. *IEEE Transactions On Systems, Man, and Cybernetics*, 22(5):885–902, September/October 1992.

[4] K. Davis. extisp.icio.us. `http://kevan.org/extispicious/`.

[5] M. Dorigo and G. D. Caro. *New Ideas in Optimization*, chapter The Ant Colony Optimization Meta-Heuristic, pages 11–32. McGraw-Hill, 1999.

[6] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, and A. Tomkins. Visualizing Tags over Time. In *Proceedings of the 15th International World Wide Web Conference (WWW 2006)*, pages 193–202, New York, NY, USA, 2006. ACM Press.

[7] A. Eaton. Graph del.icio.us related tags. `http://hublog.hubmed.org/archives/001049.html`.

[8] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[9] S. A. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[10] P. Heymann and H. Garcia-Molina. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical report, Computer Science Department, Stanford University, April 2006.

[11] A. John and D. Seligmann. Collaborative Tagging and Expertise in the Enterprise. In *Proceedings of the Collaborative Web Tagging Workshop, 15th International World Wide Web Conference (WWW 2006)*, May 2006.

[12] P. Mika. Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the 4th International Semantic Web Conference (ISWC 2005)*, November 2005.

[13] J. O'Madadhain, D. Fisher, and T. Nelson. JUNG: Java Universal Network/Graph Framework. `http://jung.sourceforge.net`.

[14] S. Pierre, O. Zitvogel, and Y. Klis. Revealicious. `http://www.ivy.fr/revealicious/`.

[15] R. C. Prim. Shortest connection networks and some generalisations. *Bell System Technical Journal*, 36, 1957.

[16] F. Viégas, S. Golder, and J. Donath. Visualizing Email Content: Portraying Relationships from Conversational Histories. In *Proceedings of the International Conference on Conference on Human Factors in Computing Systems (CHI2006)*, April 2006.

[17] S. Wasserman and K. Faust. *Social Network Analysis.* Cambridge University Press, Cambridge, 1994.

[18] Z. Xu, Y. Fu, J. Mao, and D. Su. Towards the Semantic Web: Collaborative Tag Suggestions. In *Proceedings of the Collaborative Web Tagging Workshop, 15th International World Wide Web Conference (WWW 2006)*, May 2006.

[19] Yahoo! Inc. Del.icio.us Social Bookmarking Service. `http://del.icio.us`.

[20] Yahoo! Inc. Flickr Photosharing. `http://flickr.com`.

[21] O. Zitvogel. Delicious Soup. `http://www.zitvogel.com/deliciousssoup/`.